SFC in the DOCSIS Network

James Kim
Cable Television Laboratories, Inc.

Abstract

Service Function Chaining (SFC) is a concept that has been around for a while. Newer technology advancements in the SDN and NFV space now give operators a method to implement SFC architecture in the network. SFC also gives the operators two main benefits; flexibility in offering new services to customers and data path manipulation.

CableLabs' Open Networking Project, which includes participants from cable operators, the vendor community, and CableLabs staff, has been researching SFC and trying to identify the necessary DOCSIS components to enable SFC.

This paper looks briefly at SFC and the research work currently happening in IETF and then the use cases relevant for the DOCSIS network. Finally, the paper examines different methods of introducing SFC into the DOCSIS network, identifies the benefits for each, and provides a recommendation for cable operators.

INTRODUCTION

Service Function Chaining is a dynamic working group within IETF that has made significant progress over the years. Some of the active work areas within IETF include defining the problem statement that SFC intends to address or solve, defining the SFC architecture, and identifying different use cases for SFC.

The concept of SFC is that service functions or device functions that are embedded in network devices (router, switches, etc.,) can now reside in the core or in a data-center, and can be tied into the customer's service as needed. This allows operators to offer additional services to the customer as required and permits customers to choose which services they want or do not want on their account. The SFC also allows operators to redirect customer traffic to a certain path in the SFC domain as needed. For example, a Carrier-grade Network (CGN) function can reside in the SFC domain, and operators can redirect customer traffic to the CGN function before traversing out to the Internet.

Service functions can include but are not limited to:

- Firewall
- Parental Control
- Video Optimizers
- CGN
- Traffic Engineering
- Policy Management
- Load Balancers

This list of services can be revised when new services are introduced or older or unnecessary services are no longer needed.

SFC ENCAPSULATION METHOD

Several methods exist when implementing SFC in the network. Some of the methods are using Network Service Header (NSH), Generic Network Virtualization Encapsulation (Geneve), and OpenFlow (OF).

NSH

NSH is an IETF draft. The important notion with NSH is the metadata as the metadata carries important information pertaining to that frame/packet. The metadata identifies the different service chains a frame has to traverse through. An encapsulation is added to the header to be forwarded in the SFC domain. The encapsulation ensures that the original frame will be forwarded to the SFC domain regardless of the underlying transport. Additional information regarding NSH can be found at IETF's website and also at OpenDaylight's wiki page.

GENEVE

Geneve is another method to implement SFC and, like NSH, is an IETF draft. Geneve came from network virtualization use cases where a virtual tunnel is created between two endpoints. The original intent of Geneve was to design an encapsulation technique that is flexible enough to meet the current virtualization use cases and also works with future use cases and needs. A key concept with Geneve is control plane independence from the tunnel endpoints. Another key concept is the ability to insert metadata into the frames (similar to NSH). While the creation of Geneve protocol was driven by network virtualization, the Geneve protocol can be used for SFC. Per Bruce Davie with VMware, Geneve takes the best of VXLAN, NVGRE and STT and puts it into a single protocol and encapsulation. Additional information regarding Geneve can be found at IETF's website.

OPENFLOW (OF)

Another method is using OF to implement SFC. The OF table can be used to identify packets or frames and take various actions. The OpenStack community is currently looking at using OpenFlow to enable service chain but the project is still in the incubation stage. Additional information regarding the OF specification can be found at Open Networking Foundation and information regarding the OpenStack project can be found at OpenStack's website.

Other methods of implementing SFC exist and newer methods could be developed in the near future. Each of these methods has pros and cons and will need to be reviewed in more detail at that time.

THE DOCSIS NETWORK

As stated in the introduction, SFC allows operators to easily insert or remove services. Figure 1 below shows a high-level diagram of SFC in the DOCSIS network and the data center. The two SFC edge boxes (shown in green) are the entry and exit points of the SFC domain. The various work happening within IETF and other SDOs defines the SFC domain. In Figure 1, it is assumed that there is an SDN controller for the DOCSIS network and a separate SDN controller for the data center where the SFC function resides. The orchestrator resides above the SDN controller and ensures end-to-end coordination. This model aligns with other standards, such as ETSI and MEF, looking at the orchestrator function.

Figure 1 also shows a use case in which CM1 might prefer to have Service 1, Service 2 and Service 3, whereas the CM2 would only want Service 4 and 5.

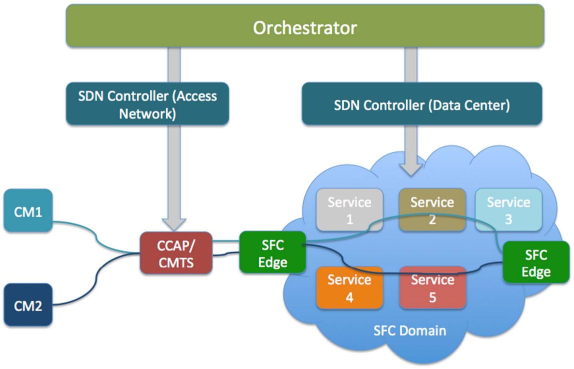

Figure 1 – Overview of SFC in the DOCSIS and Core Network

Figure 2 shows a high level description and the starting point of SFC. From the DOCSIS network, the SDN working group has identified four areas in which SFC can start. The four starting areas for SFC are the application, the home gateway, the CM and the CMTS.

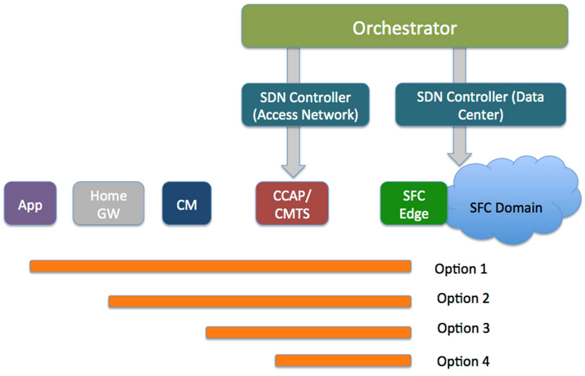

Figure 2 – Different starting points of SFC

The four options discussed next will look at SFC from the upstream direction only (from CM to CMTS). The SFC requirements for downstream have not been investigated at this time.

APPLICATION AS THE STARTING POINT

The application can be the starting point on the SFC. There are two methods from the application. The first is the overlay method which creates a tunnel (or a virtual connection) from the application to the SFC edge. In this scenario, everything from the application would be sent to the SFC. The DOCSIS network would not be aware of the application contents and there would be no classification done on the DOCSIS Network (CM/CMTS). This method would not work if the application needed to leverage the DOCSIS QoS mechanism or QoS mechanism from the CMTS to the SFC edge.

The second scenario is to mark the application so that the DOCSIS network and core network is aware. An example of this would be an MSO-provided VOIP service. The application would mark the traffic so that proper QoS could be applied on the DOCSIS network, and from the NSI side of CMTS to the SFC edge. Other MSO-provided applications, such as Video On Demand or home security, can leverage the QOS mechanism on DOCSIS and the rest of the network.

HOME GATEWAY AS THE STARTING POINT

The next starting point option is to initiate SFC from the home gateway or router. Within the home gateway option, there are several methods of implementation. The first is to use an overlay mechanism such as VXLAN or GRE from the home gateway to the SFC edge. This is similar to the first method when starting SFC from the application. The DOCSIS network and the path from the CMTS to the SFC edge would not be aware and all traffic from the home gateway would be sent to the SFC edge.

The next method is to have service chain(s) or traffic destined to the SFC edge marked and create a separate DOCSIS Service Flow. All other traffic not destined to the SFC edge would be on the primary service flow from the cable modem. This requires coordination between home gateway and the DOCSIS network so the DOCSIS network is fully aware that certain traffic is destined to the SFC edge.

CM AS THE STARTING POINT

The third option is to start at the CM. This option first appeared to be ideal, but once we started to investigate further and tried to identify the requirements from the CM, it became evident that initiating from CM was not ideal. Implementing SFC in the CM would not scale and there is no added value. The implementation that can be done on the CM can also be done on the CMTS (as described next).

CMTS AS THE STARTING POINT

The last option is to start SFC from the CMTS. Similar to the application and the home gateway, there are different methods of implementing on the CMTS. Figure 3 shows the three different methods of implementation from the CMTS.

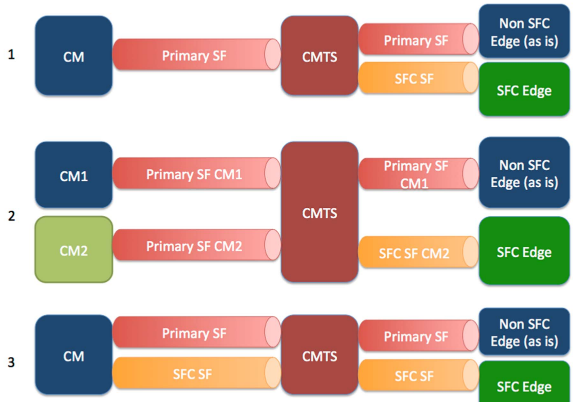

Figure 3 – Different options for CMTS

Figure 3 (CMTS option 1) shows the first method of CMTS initiating the service function. There would be one service flow in the DOCSIS network and all traffic, whether it is destined to the SFC edge or not. Once the traffic arrives at the CMTS, the CMTS would identify and separates traffic that is destined to SFC edge. This method is similar to having the CMTS perform a deep packet inspection on each frame/or packet as it arrives and leaves the CMTS.

The next method is per CM classification (as described in Figure 3, CMTS option 2). In this method, the CMTS identifies which CMs are destined for the SFC edge and which CMs are not destined for the SFC edge. Once identified, the CMTS can mark all traffic from that CM that is destined and send it to the SFC edge. The other CMs not destined for SFC edge will be routed/switched as is.

The last method (CMTS option 3) is to have multiple service flows per CM. The traffic destined for the SFC edge would be on a separate service flow and CMTS should be able to send that traffic to the SFC edge. The frames or packets not destined for SFC edge would use the primary service flow.

The first method of having the CMTS do the work in identifying the traffic type and marking will not scale. There are just too many CMs off the CMTS and each CM can have significant traffic depending on the number of users. The burden put on the CMTS might be too much at this time.

The second and third methods might be better choices, depending on the use cases and how operators want to deploy SFC. If the operator prefers to direct all traffic from the CM to the SFC domain, then the second option might be preferred. If the operator prefers to be more granular, then the third option might be preferred.

MARKING THE TRAFFIC

The above section discusses marking a customer traffic. The CMTS or the home gateway need to mark the traffic destined for the SFC edge, and the SFC edge needs the capability to read the marking. An example of marking the traffic might be to use the key field on the GRE header (if a tunneling mechanism is used) or to use the Geneve options field if using Geneve as the encapsulation method. If an operator is running an MPLS network, an MPLS shim header can also be used between the PE and the SFC edge.

RECOMMENDATION

Each starting point has pros and cons. The application starting point is very granular and there is doubt about whether the traffic should go through the SFC domain or not. Starting SFC from the application is useful when the operator has full control over the application. The downside to this method is that it is limited at this point; however, as MSOs start to expand into the home network, using an overlay technique may be the easiest to implement. Unfortunately, it does not take full advantage of the DOCSIS network and the rest of the MSO's network. Further research is needed where the applications leverage the DOCSIS and the core network, not the preferred network.

As stated before, starting from the CM doesn't scale. Legacy CMs might not support it, and using it does not offer any advantages.

Finally, a combination of using the home gateway and the CMTS might be the best option available right now. More research is needed to identify what is needed between the home gateway and the CMTS, but this option allows SFC to take advantage of the DOCSIS network and the core network.

REFERENCES

1. Service Function Chaining (SFC) Architecture, draft-ietf-sfc-architecture-07, https://datatracker.ietf.org/doc/draft-ietf-sfc-architecture/

2. Network Service Header, draft-quinn-sfc-nsh-07, https://datatracker.ietf.org/doc/draft-quinn-sfc-nsh/

3. VM Ware Blog on Geneve, VXLAN and Network Virtualization Encapsulations, http://blogs.vmware.com/cto/geneve-vxlan-network-virtualization-encapsulations/

4. Geneve: Generic Network virtualization Encapsulation, draft-gross-geneve-02, https://datatracker.ietf.org/doc/draft-gross-geneve/

5. IETF RFC 2784, Generic Routing Encapsulation.

6. OpenStack Neutron, Service Function Chaining using OpenFlow, https://blueprints.launchpad.net/neutron/+spec/service-function-chaining-using-openflow